



Prioritätsbescheinigung über die Einreichung einer Patentanmeldung

Aktenzeichen: 101 25 385.0

Anmeldetag: 23. Mai 2001

Anmelder/Inhaber: Siemens AG, München/DE

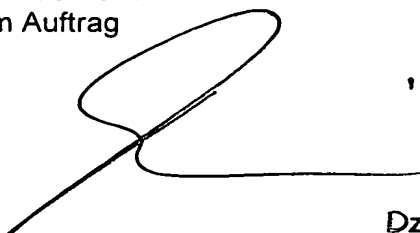
Bezeichnung: Vorrichtung und Verfahren zur Statusprüfung von Steuerungssystemen

Priorität: 18.12.2000 DE 100 63 044.8

IPC: G 06 F 11/36

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 13. November 2001
Deutsches Patent- und Markenamt
Der Präsident
Im Auftrag



Dzierzon

Beschreibung

Vorrichtung und Verfahren zur Statusprüfung von Steuerungssystemen

5

Die vorliegende Erfindung betrifft eine Vorrichtung zum Parametrieren, Projektieren und Inbetriebnehmen von Steuerungssystemen und/oder zum Erstellen von vorzugsweise zyklischen Steuerungsprogrammen mit einer Editoreinrichtung zum Editieren eines Steuerungsprogramms und einer Compilereinrichtung zum Kompilieren des Steuerungsprogramms. Darüber hinaus be-
10 trifft die vorliegende Erfindung eine Vorrichtung zur zeitlichen Steuerung eines Systems auf der Grundlage eines Steuerungsprogramms mit einer Mikroprozessoreinrichtung zur Abar-
15 beitung eines kompilierten, vorzugsweise zyklischen Steuerungsprogramms. Ferner wird eine Anordnung aus beiden Vorrichtungen beschrieben.

20

Bei speicherprogrammierbaren Steuerungen wird vielfach ein so genanntes Engineeringsystem dazu verwendet, um ein Steuerungssystem zu debuggen, parametrieren und in Betrieb zu nehmen, sowie geeignete Steuerungsprogramme zu erstellen. Des Weiteren wird ein so genanntes Runtime-System verwendet, um die Daten des Engineeringsystems aufzunehmen und die Steuerungsprogramme entsprechend abzuarbeiten.

25

Das Debuggen von Steuerungsprogrammen und Beobachten von Steuerungsvariablen ist bei derzeit eingesetzten Systemen kaum oder nur sehr unkomfortabel machbar. Bei derzeit gängi-
30 gen Technologien ist es notwendig, dass eine entsprechende Hardware-Unterstützung eingerichtet ist. Dies bedeutet, dass beispielsweise nur über bestimmte Breakpoint-Mechanismen oder Interrupts von Prozessoren das Debugging möglich ist. Bei interpretativ arbeitenden Steuerungen wird das Debugging bis-
35 lang interpretativ gelöst, was aber sehr starke Laufzeiteinflüsse hat.

Insbesondere bei zyklisch ablaufenden Steuerungsprogrammen ist das Beobachten von Variablen mit bisherigen Mitteln kaum möglich. Der Anwender kann das Programm am Bildschirm nicht so rasch verfolgen wie ein SPS-Zyklus (speicherprogrammierbare Steuerung) läuft. Die Beobachtung im SPS-Zyklus belastet außerdem die Runtime-CPU verhältnismäßig stark und wäre für den Anwender wenig hilfreich.

Angesichts dessen besteht die Aufgabe der vorliegenden Erfindung darin, eine Vorrichtung und ein Verfahren vorzuschlagen, mit denen das Debuggen und das Beobachten von Systemparametern während des Laufs eines Steuerungsprogramms komfortabler durchführbar ist.

Erfindungsgemäß wird diese Aufgabe gelöst durch eine Vorrichtung zum Parametrieren, Projektieren und Inbetriebnehmen von Steuerungssystemen und/oder zum Erstellen von vorzugsweise zyklischen Steuerungsprogrammen mit einer Editoreinrichtung zum Editieren eines Steuerungsprogramms und einer Compiler-einrichtung zum Kompilieren des Steuerungsprogramms, wobei mit der Editoreinrichtung der gesamte oder ein Teilbereich des Steuerungsprogramms zum Debuggen markierbar ist oder das gesamte Steuerungsprogramm zum Debuggen instrumentierbar ist und mit der Compilereinrichtung aus dem Steuerungsprogramm ein Zwischencode erzeugbar ist, der für den markierten Bereich des Steuerungsprogramms und/oder für das gesamte Steuerungsprogramm Debugginstrumentierung enthält.

Ferner wird die oben genannte Aufgabe gelöst durch eine Vorrichtung zur zeitlichen Steuerung eines Systems auf der Grundlage eines Steuerungsprogramms mit einer Mikroprozessoreinrichtung zur Abarbeitung eines kompilierten, vorzugsweise zyklischen Steuerungsprogramms und einer Compilereinrichtung zum Kompilieren eines vorkompilierten Steuerungsprogramms mit Debugginstrumentierung in einen Objektcode für die Mikroprozessoreinrichtung.

Schließlich wird die genannte Aufgabe durch Verfahren gelöst, die sich aus den obigen Vorrichtungen ergeben.

5 In vorteilhafter Weise ist der Zwischencode unabhängig von einem Mikrocontrollertyp. Bei einer Weiterentwicklung kann eine Datenhaltungseinrichtung für Zuordnungsinformation zum Zuordnen des markierten Bereichs des Steuerungsprogramms zu einem Befehl des Zwischencodes vorgesehen sein. Ferner kann die Editoreinrichtung eine Auftragseinheit zum Absenden eines
10 Beobachtungsauftrags hinsichtlich des markierten Bereichs enthalten. Entsprechend kann die Editoreinrichtung eine Empfangseinrichtung zum Empfangen von Beobachtungsinformationen umfassen.

15 Die Vorrichtung zur zeitlichen Steuerung eines Systems auf der Grundlage eines Steuerungsprogramms kann mit einem Beobachtungsmodul, das anhand der Debugginstrumentierung einrichtbar ist, ausgestattet sein. Ferner kann eine Datenpuffereinrichtung zum Speichern und Bereitstellen von Beobach-
20 tungsinformationen von dem Beobachtungsmodul vorgesehen sein. Mit einer Empfangseinrichtung kann dann die Vorrichtung einen Abarbeitungsauftrag für das Beobachtungsmodul empfangen.

25 Als besonders vorteilhaft erweist sich eine Anordnung aus den beiden genannten Vorrichtungen zum Steuern eines Systems.

Die vorliegende Erfindung wird nun anhand der beigefügten Zeichnungen näher erläutert, in den zeigen:

30 Fig. 1 ein Datenflussdiagramm einer erfindungsgemäßen Anordnung; und

Fig. 2 ein Datenflussdiagramm einer Statusabfrage.

35 Das nachfolgende Ausführungsbeispiel dient dazu, die vorliegende Erfindung besser vertraut zu machen.

Gemäß Fig. 1 wird in dem Engineeringsystem ES ein Steuerungsprogramm 1, beispielsweise im structured text-Format, hinterlegt. Für das Steuerungsprogramm 1 wird durch einen Compiler 2 ein speziell instrumentierter Zwischencode 3 erzeugt. Dieser Zwischencode 3 enthält zusätzliche Informationen mit deren Hilfe später ein Runtime-System RS Statusabfragen über entsprechende Variablen ermöglicht. Der Compiler 2 kompiliert das Steuerungsprogramm vor dem Download in das Runtime-System RS. In dem Runtime-System RS bzw. in der Steuerung wird dieser Zwischencode 3 durch einen weiteren Compiler 4 in einen mikroprozessorspezifischen Code 5 umgesetzt.

Die Compiler-Funktionalität, den Sourcecode des Steuerungsprogramms 1 in einen maschinenunabhängigen Zwischencode umzusetzen ist in dem Basis-Engineeringsystem ES realisiert. Das Basis-Engineeringsystem ES besitzt eine spezifische Datenhaltung 6 für Debug- bzw. Zuordnungsinformation. Diese Zuordnungsinformation gibt an, welche Steuerungsprogrammzeilen welchen Zwischencodeabschnitten entsprechen.

Gemäß Fig. 2 kann der Anwender durch graphische Markierungen 7 eine Abschnittskennung von Steuerungsprogrammzeilen vornehmen. Darauf aufsetzend ermittelt ein Debugger 8 des Engineeringsystems den dazugehörigen Zwischencodebereich, wobei die entsprechenden Bereichsangaben in der Datenhaltung 6 hinterlegt werden. Die Zuordnung kann z. B. über eine Zuordnungstabelle oder einen Verweis erfolgen.

Auf der Seite des Engineeringsystems steht somit ein Steuerungsprogramm 1 mit einem kompilierten Zwischencode 3. Dieser Zwischencode 3, der instrumentiert ist, wird dann ins Runtime-System RS geladen. Das Runtime-System RS setzt nicht nur den Zwischencode 3 in den Mikroprozessorcode 5 um, sondern verfügt auch über eine eigene Debug-Funktionalität 9 mit dem anhand des dem markierten Bereich des Steuerungsprogramms entsprechenden Zwischencodeabschnitts in einer nicht darge-

stellten Datenhaltung der Bereich 10 des Mikroprozessorcodes 5 ermittelt und gegebenenfalls beobachtet wird.

Das Steuerungsprogramm kann nun abgearbeitet werden, d. h.
5 die CPU des Runtimesystems RS kann in run gehen. Der Mikro-
prozessorcode wird zyklisch durchlaufen. Da nun der Anwender
im Steuerungsprogramm einen bestimmten Bereich markiert hat,
wird ein Auftrag an das RuntimeSystem RS generiert, welcher
Zwischencodebereich beobachtet werden soll. Das RuntimeSystem
10 RS ermittelt die Mikroprozessorcodebereiche aus dieser Infor-
mation und richtet die Infrastruktur für eine Programmbe-
obachtung ein. Dies bedeutet, dass spezielle Puffer bereitge-
stellt werden, um Statusinformationen zwischenzuspeichern.
Nachdem dieser Dienst eingerichtet wurde, kann er vom Anwen-
15 der potentiell verwendet werden.

Wird nun die Programmbeobachtungsfunktion vom Anwender expli-
zit, z. B. im Editor, angestoßen, so wird im RuntimeSystem RS
ein Beobachtungsmodul eingerichtet. Wenn das Programm in ei-
20 nen dieser Maschinencodebereiche, die dem markierten Bereich
des Steuerungsprogramms entsprechen, eintritt, so werden die
betroffenen Variablen Werte in entsprechende Datenpuffer 11
geschrieben. Dies erfolgt solange wie sich das Programm in
diesen speziellen Codebereichen befindet.

25 Wird der Score des Codebereichs verlassen, bekommt das Engi-
neeringsystem ES von diesem Beobachtungsmodul eine Notifika-
tion. Das Engineeringsystem holt sich die Informationen aus
dem Datenpuffer 11. Das Debug-Modul 8 im Engineeringsystem
30 ermittelt aus der Zwischencodezuordnung und diesem Datenpuf-
fer 11 die entsprechende Zeileninformation für das Steu-
erungsprogramm. Daraufhin können die vom Anwender gewünschten
Variablen angezeigt werden. Ist die Anzeige erfolgt, so kann
auf Wunsch ein neuer Beobachtungsauftrag für diese eingerich-
35 tete Infrastruktur abgesetzt werden. Dies bedeutet, dass der
Beobachtungs- bzw. Debug-Zyklus oder die Flusssteuerung ei-
gentlich über den Editor erfolgt. Dies ist insofern sinnvoll,

da der Anwender das Programm am Bildschirm nicht im Takt des SPS-Zyklus verfolgen kann. Darüber hinaus würde die Beobachtung im SPS-Zyklus die Runtime-CPU übermäßig belasten. Die Daten der abgefragten Variablen werden also nur auf expliziten Wunsch oder beispielsweise periodisch jede Sekunde aktualisiert.

Aktiviert der Anwender den Debug-Mechanismus 8 im Engineersystem ES, so werden ihm in einem zweiten Fenster 12 neben seiner Steuerprogrammquelle 1 für den entsprechend zyklisch durchlaufenen Programmcode die Werte der im Programmcode verwendeten Variablen konsistent für den jeweiligen Durchlauf angezeigt. Damit kann der Anwender komfortabel eine Diagnose und einen Programm-Debug bei zyklischer SPS-Funktionalität realisieren. Für eine Realisierung des Mechanismus ist eine interpretative Lösung aus Performancegründen weniger vorteilhaft.

Der Vorteil der vorliegenden Erfindung besteht somit darin, dass das Debuggen bzw. Beobachten von Variablen hardware-unabhängig erfolgt. Darüber hinaus wird durch die erfindungsgemäße Technik die CPU des Runtimesystems nur so stark belastet, wie der Programmeditor diese Beobachtung ausnutzen kann. Dies bedeutet, dass die Programmlaufzeit allenfalls durch das Beobachten des markierten Bereichs verzögert wird. Die Ursache hierfür liegt darin, dass nur für den markierten Bereich des Steuerungsprogramms der Debug-Code durchlaufen wird. Alle anderen Bereiche werden ohne Debug-Option abgearbeitet.

Wie bereits angedeutet eignet sich die erfindungsgemäße Funktionalität und dieser Mechanismus des selektiven Debuggens bestens zum Debuggen zyklisch laufender Programme. Aber auch Motion Tasks, die nicht zyklisch laufen, können sehr gut beobachtet werden. Insbesondere kann durch einen zusätzlichen Mechanismus der Tracebuffer 11 schon vor Verlassen des Scopes abgeholt werden. Man kann dadurch z. B. ermitteln, wann man

mit welchen Parametern in synchrone Aufrufe gerät. Auch zum Auffinden von Deadlocks in Motion Tasks eignet sich das geschilderte Verfahren und der dazugehörige Mechanismus.

Patentansprüche

- 5 1. Vorrichtung zum Parametrieren, Projektieren und Inbetriebnehmen von Steuerungssystemen und/oder zum Erstellen von vorzugsweise zyklischen Steuerungsprogrammen mit
- 10 einer Editoreinrichtung zum Editieren eines Steuerungsprogramms (1) und
- einer Compilereinrichtung (2) zum Kompilieren des Steuerungsprogramms (1),
- 15 dadurch gekennzeichnet, dass
- mit der Editoreinrichtung der gesamte oder ein Teilbereich (7) des Steuerungsprogramms (1) zum Debuggen markierbar ist und
- 20 mit der Compilereinrichtung (2) aus dem Steuerungsprogramm (1) ein Zwischencode (3) erzeugbar ist, der für den markierten Bereich (7) des Steuerungsprogramms (1) Debuginstrumentierung enthält.
- 25 2. Vorrichtung nach Anspruch 1, wobei der Zwischencode (3) unabhängig von einem Mikrokontrollertyp ist.
- 30 3. Vorrichtung nach Anspruch 1 oder 2, wobei eine Dateneinrichtung (6) für Zuordnungsinformation zum Zuordnen des markierten Bereichs (7) des Steuerungsprogramms (1) zu einem Bereich des Zwischencodes (3) vorgesehen ist.
- 35 4. Vorrichtung nach Anspruch 1, 2 oder 3, wobei die Editoreinrichtung eine Auftragseinheit zum Absenden eines Beobachtungsauftrags hinsichtlich des markierten Be-

reichs umfasst.

5 5. Vorrichtung nach einem der Ansprüche 1 bis 4, wobei die Editoreinrichtung eine Empfangseinrichtung zum Empfangen von Beobachtungsinformation (12) umfasst.

6. Vorrichtung zur zeitlichen Steuerung eines Systems auf der Grundlage eines Steuerungsprogramms (1) mit

10 einer Mikroprozessoreinrichtung zur Abarbeitung eines kompilierten, vorzugsweise zyklischen Steuerungsprogramms (5),

gekennzeichnet durch

15

eine Compilereinrichtung (4) zum Kompilieren eines vor-kompilierten Steuerungsprogramms (3) mit Debuginstrumentierung in einen Objektcode (5) für die Mikroprozessoreinrichtung.

20

7. Vorrichtung nach Anspruch 6 mit einem Beobachtungsmodul (9, 10, 11), das anhand der Debuginstrumentierung einrichtbar ist.

25

8. Vorrichtung nach Anspruch 7 mit einer Datenpuffereinrichtung (11) zum Speichern und Bereitstellen von Beobachtungsinformation von dem Beobachtungsmodul.

30

9. Vorrichtung nach Anspruch 7 oder 8 mit Empfangseinrichtung zum Empfangen eines Abarbeitungsauftrags für das Beobachtungsmodul.

10. Anordnung zum Steuern eines Systems mit

35

einer Vorrichtung zum Parametrieren, Projektieren und Inbetriebnehmen von Steuerungssystemen und/oder zum Erstellen von vorzugsweise zyklischen Steuerungspro-

grammen (1) nach einem der Ansprüche 1 bis 5 und

einer Vorrichtung zur zeitlichen Steuerung eines Systems auf der Grundlage eines Steuerungsprogramms nach einem der Ansprüche 6 bis 9, wobei

der Zwischencode (3) dem vorkompilierten Steuerungsprogramm entspricht.

10 11. Verfahren zum Parametrieren, Projektieren und Inbetriebnehmen von Steuerungssystemen und/oder zum Erstellen von vorzugsweise zyklischen Steuerungsprogrammen (1) durch

15 Editieren eines Steuerungsprogramms (1) und

Kompilieren des Steuerungsprogramms (1),

gekennzeichnet durch

20 Markieren des gesamten Bereichs oder eines Teilbereichs (7) des Steuerungsprogramms zum Debuggen beim Editieren und

25 Erzeugen ein Zwischencodes (3) aus dem Steuerungsprogramm (1) beim Kompilieren, der für den markierten Bereich (7) des Steuerungsprogramms Debuginstrumentierung enthält.

30 12. Verfahren nach Anspruch 11, wobei der Zwischencode (3) unabhängig von einem Mikrokontrollertyp ist.

13. Verfahren nach Anspruch 11 oder 12, wobei der markierte Bereich (7) des Steuerungsprogramms (1) einem Bereich
35 des Zwischencodes (3) zugeordnet wird.

11

14.Verfahren nach Anspruch 11, 12 oder 13, wobei beim Editieren ein Beobachtungsauftrag zum Beobachten des markierten Bereichs (7) abgesendet wird.

5 15.Verfahren nach einem der Ansprüche 11 bis 14, wobei beim Editieren Beobachtungsinformation (12) empfangen wird.

10 16.Verfahren zur zeitlichen Steuerung eines Systems auf der Grundlage eines Steuerungsprogramms (1) durch

Abarbeitung eines kompilierten, vorzugsweise zyklischen Steuerungsprogramms (5),

15 gekennzeichnet durch

Kompilieren eines vorkompilierten Steuerungsprogramms (3) mit Debuginstrumentierung in einen Objektcode (5) für eine Mikroprozessoreinrichtung.

20

17.Verfahren nach Anspruch 16, wobei ein Beobachtungsmodul (9, 10, 11) anhand der Debuginstrumentierung eingerichtet wird.

25 18.Verfahren nach Anspruch 17, wobei Beobachtungsinformation (12) von dem Beobachtungsmodul in einer Datenpuffereinrichtung (11) gespeichert und bereitgestellt wird.

30 19.Verfahren nach Anspruch 17 oder 18, wobei während der Abarbeitung des Steuerungsprogramms (5) ein Abarbeitungsauftrag für das Beobachtungsmodul empfangen und ausgeführt wird.

35 20.Verfahren zum Steuern eines Systems mit

einem Verfahren zum Parametrieren, Projektieren und In-

12

betriebsnehmen von Steuerungssystemen und/oder zum Erstellen von vorzugsweise zyklischen Steuerungsprogrammen nach einem der Ansprüche 11 bis 15 und

5 einem Verfahren zur zeitlichen Steuerung eines Systems auf der Grundlage eines Steuerungsprogramms nach einem der Ansprüche 16 bis 19, wobei

10 der Zwischencode (3) dem vorkompilierten Steuerungsprogramm entspricht.

Zusammenfassung

Vorrichtung und Verfahren zur Statusprüfung von Steuerungssystemen

5

Das Debuggen zyklisch ablaufender Steuerungsprogramme soll in einem Steuerungssystem komfortabler gestaltet werden. Dazu wird das Steuerungsprogramm (1) in einen mikroprozessorunabhängigen Zwischencode (3) kompiliert. Dieser Zwischencode (3) wird in das Runtime-System RS geladen und dort in den Mikroprozessorcode (5) kompiliert. Bereichsangaben über zu untersuchende Codebereiche werden in einer eigenen Datenhaltung (6) hinterlegt und mittels Debug-Funktionalität im Engineering-System und im Runtime-System auf Bereiche des Maschinencodes abgebildet.

15

(Fig. 1)

1/2

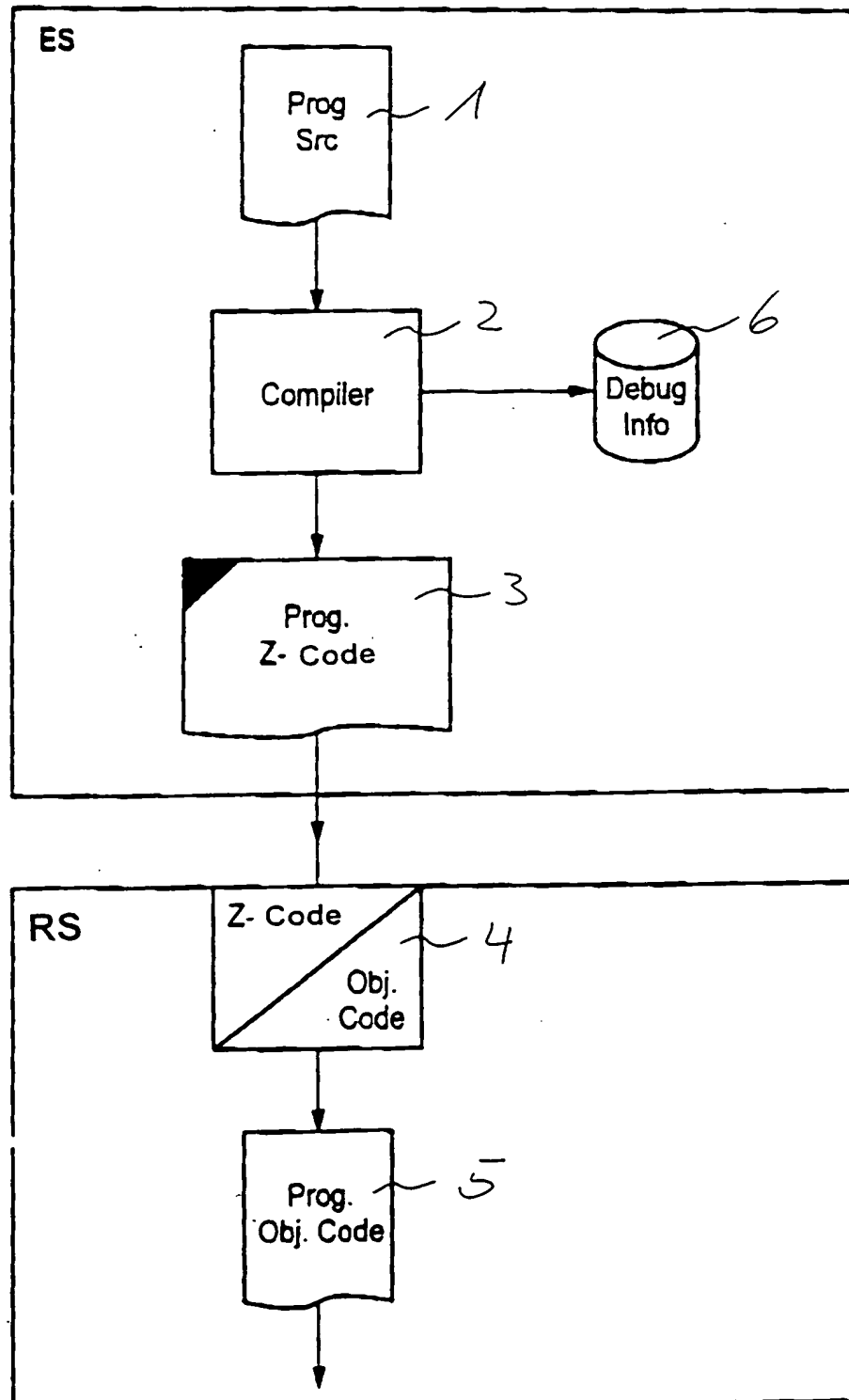


FIG 1

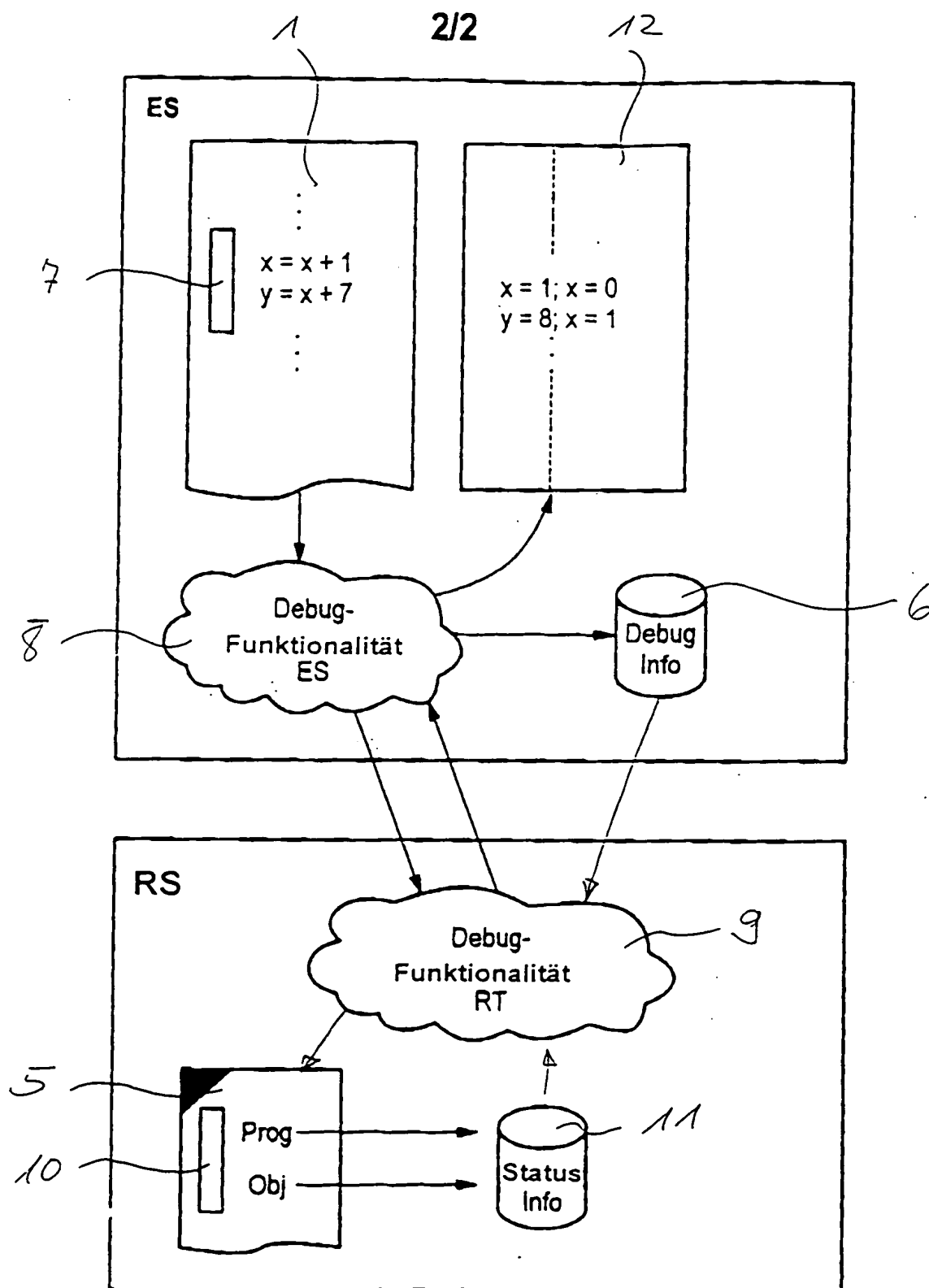


FIG 2